Graph concepts

Artificial intelligence (CK0031)

Francesco Corona

Graphs

Usually we have good reason to believe that one event affects another

- or conversely that some events are independent

Incorporating such knowledge can yield models that are better specified

- and computationally more efficient

Graphs describe how objects are linked and provide
a convenient picture for describing related objects

Remark

We will ultimately introduce a graph structure among the variables of a probabilistic model to produce a 'probabilistic graphical model'

- We want to capture relations among variables as well as their uncertainties

Graphs

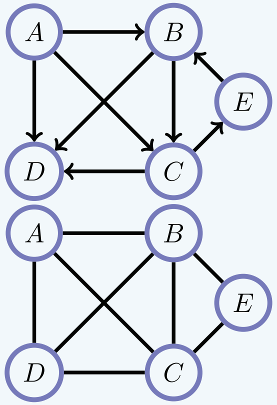
Graph concepts

Definition

Graphs

A **graph** $\mathcal{K} = (\mathcal{A}, \mathcal{E})$ is a data structure consisting of a set of **nodes** $\mathcal{A} = \{A_1, \ldots, A_N\}$ and a set of **edges** $\mathcal{E}$ between pairs of nodes in $\mathcal{A}$

- Edges may be directed ($A_i \rightarrow A_j$) or undirected ($A_i - A_j$)
- Edges can have associated weights



- **Directed graph** $\mathcal{G}$: All edges are directed ($A_i \rightarrow A_j$ or $A_j \rightarrow A_i$)
- **Undirected graph** $\mathcal{H}$: All edges are undirected ($A_i - A_j$)

Graph concept (cont.)

Our use of graphs is to endow them with a probabilistic interpretation and we develop a connection between directed graphs and probability

Undirected graphs are central in modelling/reasoning with uncertainty

- Variables are independent if not linked by a path on the graph

Graphs concepts (cont.)

Definition

Walks, **trails** and **paths**

A **walk** $A \mapsto B$ from node A to node B is an alternating sequence of nodes and edges that connects A and B

- A walk is of a form $A_0, e_1, A_1, e_2 \dots, A_{M-1}, e_M, A_M$ with $A_0 = A$ and $A_M = B$ and each edge (A_{m-1}, A_m) with $m = 1, \dots, M$ being in the graph, M is said to be the **length** of the walk
- A directed graph is a sequence of nodes which, when we follow the direction of the arrows, leads us from A to B

Refinement of a walk

- **Trails**, which are walks without repeated edges
- **Paths**, which are trails without repeated nodes

Graphs concepts (cont.)

Definition

Ancestors and **descendants**, **parents** and **children**

In directed graphs

- Nodes A, such that $A \mapsto B$ and $B \not\mapsto A$ are the **ancestors** of B
- Nodes B, such that $A \mapsto B$ and $B \not\mapsto A$ are the **descendants** of A

Wherever we have that $A_i \rightarrow A_j \in \mathcal{E}$, we say that

- A_j is the **child** of A_i in $\mathcal{K}$, $\text{ch}(A_j)$ denotes the children of A_j
- A_i is the **parent** of A_j in $\mathcal{K}$, $\text{pa}(A_i)$ denotes the parents of A_i

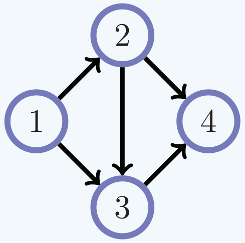
Graph concepts (cont.)

Definition

Cycles and **loops**

A **cycle** is a directed path that starts and returns to the same node

$$a \rightarrow b \rightarrow \cdots \rightarrow z \rightarrow a$$



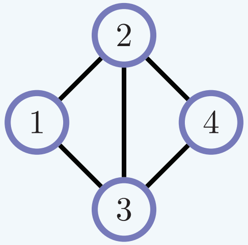
A **loop** is a path containing more than two nodes, irrespective of edge direction, that starts and returns to the same node

- $1 - 2 - 4 - 3 - 1$ forms a loop
- The graph is **acyclic**

Graph concepts (cont.)

Definition

Chords



Adjacency is a notion of connectivity

- Two nodes A_i and A_j are said to be **adjacent** if joined by an edge in $\mathcal{E}$

A **chord** is an edge that connects two non-adjacent nodes in a loop

- Edge $2-3$ is a chord in the $1-2-4-3-1$ loop

Graph concepts (cont.)

Definition

Directed Acyclic Graph, **DAG**

A **DAG** is a graph $\mathcal{G}$ with directed edges between the nodes such that

- by following a path of nodes from one node to another along the direction of each edge no path will revisit a node

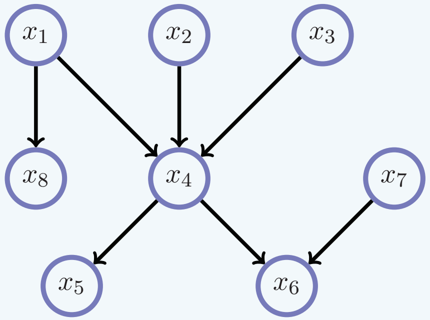
In a DAG:

- Ancestors of B are nodes who have a directed path ending at B
- Descendants of A are nodes who have a directed path starting at A

Graph concepts (cont.)

Definition

Relations in a DAG



- The **parents** of x_4

  $$\text{pa}(x_4) = \{x_1, x_2, x_3\}$$

- The **children** of x_4

  $$\text{ch}(x_4) = \{x_5, x_6\}$$

The **Markov blanket** of a node is its parents, its children and the parents of its children (excluding itself)

- The Markov blanket of x_4 is $\{x_1, x_2, x_3, x_5, x_6, x_7\}$

Graph concepts (cont.)

Remark

One can view directed links on a graph as ‘direct dependencies’ between parent and child, with acyclicity condition preventing circular reasoning

Graph concepts (cont.)

Definition

Neighbours and **boundary**

For an undirected graph $\mathcal{G}$, the **neighbours** of x, ne(x), are those nodes directly connected to x

We define the **boundary** of x, boundary(x), to be pa(x) $\cup$ ne(x)

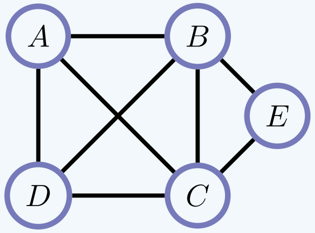
Graph concepts (cont.)

Definition

Clique

Given a graph, a **clique** is a fully connected (complete) subset of nodes

- All the member of the clique are neighbours
- For the **maximal clique**, no larger clique containing the clique



Two maximal cliques

- $\mathcal{C}_1 = \{A, B, C, D\}$
- $\mathcal{C}_2 = \{B, C, E\}$

Whilst A, B, C are fully connected, this is a non-maximal clique, it is a **cliquo**, as $\{A, B, C, D\}$ is a larger fully connected set that contains this

Graph concept (cont.)

Cliques play a central role in both modelling and inference

- In modelling, they describe variables that are all dependent on each other
- In inference, they describe sets of variables with no simpler structure describing the relationship between them (for which no simpler efficient inference procedure is likely to exist)

Graphs concepts (cont.)

Definition

Connected graph

An undirected graph is said to be **connected** if there is a path between every pair of nodes

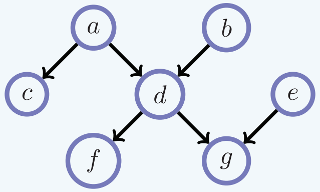
- There are no isolated islands (uh?!)

For a non-connected graph, the **connected components** are those subgraphs which are connected

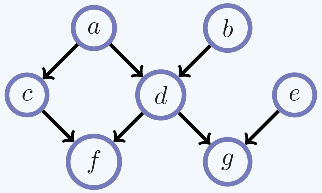
Definition

Singly-connected graph

A graph is **singly connected** if there is only one path from any node A to any other node B, otherwise the graph is **multiply connected**



Singly-connected graph
It is also called a **tree**



Multiply-connected graph
It is also called **loopy**

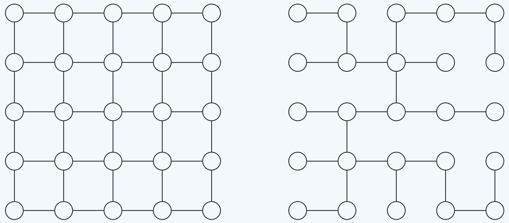
Definition applies regardless of whether or not the edges are directed

Graph concepts (cont.)

Definition

Spanning tree

A **spanning tree** of an undirected graph $\mathcal{G}$ is a singly-connected subset of edges st the resulting singly-connected graph covers all nodes of $\mathcal{G}$



A **maximum weight spanning tree** is a spanning tree st the sum of all weights on the edges of the tree is at least as large as any spanning tree

Graph concepts (cont.)

Pseudocode

Finding a maximal weight spanning tree

An algorithm to find a spanning tree with maximal weight is as follows

1. Pick the edge with the largest weight and add it to the edge set
2. Pick the next candidate edge and add it to the edge set
3. If this results in an edge set with cycles, reject the candidate edge and propose the next largest edge weight

Note that there may be more than one maximal weight spanning tree

Numerical encoding

Graph concepts

Numerical encoding

Our prime goal is to make computational implementations of inference

If we want to incorporate graph structure into our models, we need to express graphs in a way that a computer can understand and manipulate

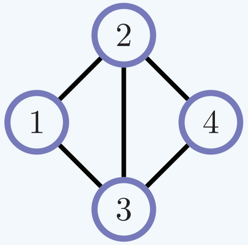
There are several equivalent possibilities

Edge list

Definition

An **edge list** is a list containing which node-node pairs are in the graph

$L = \{(1,2),(2,1),(1,3),(3,1),(2,3),(3,2),(2,4),(4,2),(3,4),(4,3)\}$



Undirected edges are listed twice

- once for each direction

Adjacency matrix

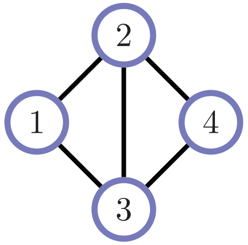
Definition

An alternative: The $|\mathcal{A}| \times |\mathcal{A}|$ binary matrix $\mathbf{A}$ called **adjacency matrix**

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \tag{1}$$

$A_{ij} = 1$ if there is an edge from node i to node j, and $A_{ij} = 0$ otherwise

- some include self-connections 1s on the diagonal ($A_{ij} = 1$, for $i = j$)



An undirected graph has a symmetric adjacency matrix

Adjacency matrix (cont.)

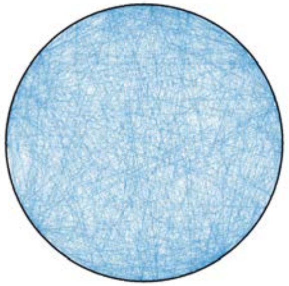
Adjacency matrices are useful not only for storing connectivity info

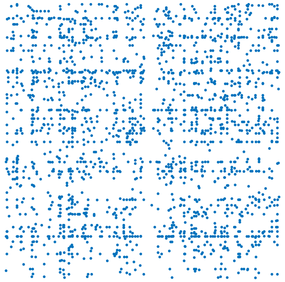
- Certain operations on $\mathbf{A}$ yield additional info concerning $\mathcal{G}$

The row-sum $\mathbf{A}_{i+} = \sum_j A_{ij}$ is equal to the **degree** d_i of node i

- The **degree** of a node x is the number of edges incident on the node
- A node x is **incident** on an edge e, if x is an endpoint of e

Note that, by symmetry, $\mathbf{A}_{i+} = \mathbf{A}_{+i}$

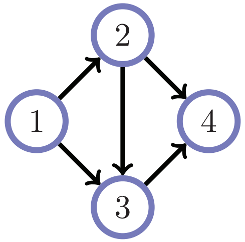




Adjacency matrix (cont.)

If nodes are labelled in **ancestral order** (parents always before children) a directed graph can be represented as a triangular adjacency matrix

$$\mathbf{T} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{2}$$



A directed graph with nodes labelled in ancestral order corresponds to a triangular adjacency matrix

Adjacency matrix (cont.)

Adjacency matrices may seem wasteful since many entries are zero

- However, they have a useful property

Definition

For a $N \times N$ adjacency matrix $\mathbf{A}$, powers of the adjacency matrix $[\mathbf{A}^k]_{ij}$ specify how many paths there are from node i to node j in k edge hops

- If we include 1s on the diagonal of $\mathbf{A}$ then $[\mathbf{A}^{N-1}]_{ij}$ is non-zero when there is a path connecting i to j in the graph
- If $\mathbf{A}$ corresponds to a DAG the non-zero entries of the j-th row of $[\mathbf{A}^{N-1}]$ corresponds to a descendant of node j

Adjacency matrix (cont.)

Exercise

Consider an adjacency matrix $\mathbf{A}$, with elemnents $[\mathbf{A}]_{ij} = 1$ if one can reach state i from state j in one time step, and 0 otherwise

- Show that the matrix $[\mathbf{A}^k]_{ij}$ represents the number of paths that lead from state j to state i in k steps
- Derive an algorithm that will find the minimum number for steps to get from state j to state i

Incidence matrix and graph Laplacian

Definition

Incidence matrix: $\mathbf{B}$, $|\mathcal{A}| \times |\mathcal{E}|$ binary matrix capturing structure in $\mathcal{G}$

$$B_{ij} = \begin{cases} 1, & \text{if vertex } i \text{ is incident to edge } j \\ 0, & \text{otherwise} \end{cases}$$

If we extend $\mathbf{B}$ to a signed incidence matrix $\tilde{\mathbf{B}}$ in which entries 1 of $\mathbf{B}$ are given a $+$ or a $-$ sign indicating an arbitrarily assigned *orientation* of the corresponding edge, then it can be shown that $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^T = \mathbf{D} - \mathbf{A} = \mathbf{L}$

$\mathbf{D} = \text{diag}\big[(d_i)_{i \in \mathcal{V}}\big]$ is a diagonal matrix with the degree sequence

$\mathbf{L}$ is the $|\mathcal{V}| \times |\mathcal{V}|$ **graph Laplacian** of $\mathcal{G}$

For a $\mathbf{x} \in \mathbb{R}^{|\mathcal{V}|}$, we have $\mathbf{x}^T\mathbf{L}\mathbf{x} = \sum_{\{i,j\} \in \mathcal{E}} (x_i - x_j)^2$, which gets closer to zero as the elements of $\mathbf{x}$ at adjacent nodes in $\mathcal{V}$ get more similar

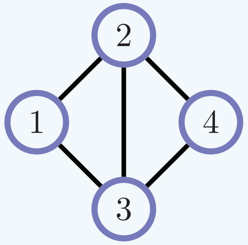
- It can be understood as a measure of *smoothness* of functions on $\mathcal{G}$, with respect to its connectivity

Clique matrix

Definition

For an undirected graph with N nodes and maximal cliques $\mathcal{C}_1, \dots, \mathcal{C}_k$

- a **clique matrix** is a $N \times K$ matrix in which each column c_k has zeros except for ones on entries describing the clique



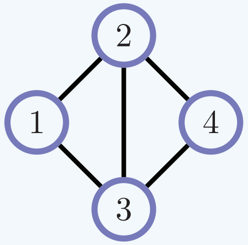
$$\mathbf{C} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad (3)$$

A **cliquo matrix** relaxes the constraint that cliques need be maximal

Clique matrix (cont.)

Definition

A cliquo matrix containing only two-node cliques is an **incidence matrix**



$$\mathbf{C}_{\text{inc}} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \qquad (4)$$

$\mathbf{C}_{\text{inc}}\mathbf{C}_{\text{inc}}^T$ is equal to the adjacency matrix, except that the diagonals contain the **degree** of each node (the number of nodes it touches)

- For any cliquo matrix, the diagonal entry of $[\mathbf{C}\mathbf{C}^T]_{ii}$ expresses the number of cliquos (columns) that node i occurs in
- Off-diagonal elements $[\mathbf{C}\mathbf{C}^T]_{ij}$ contain the number of cliquos that node i and j jointly inhabit